2 Least mean squares via Gradient descent

2.0.1 Gradient descent

The gradient descent algorithm finds a local minima of the objective function (J) by guessing an initial set of parameters w and then "walking" episodically in the direction of the gradient $\partial J/\partial w$. Since w is vector valued (3 components for our example) we need to perform the update for each component separately

$$w^j = w^j - \lambda \frac{\partial J(w)}{\partial w^j}$$

where λ is the *learning rate* parameter or the step of the update. To see this in practice, lets consider an example (other than the mean square error): say $J(x) = (x-2)^2 + 1$ and the initial guess for a minimum is $x_0 = 3$. The differential of J is

$$\frac{\partial J(x)}{\partial x} = 2x - 4$$

Assuming a fixed $\lambda = 0.25$, the first several episodes of gradient descent are:

$$x_1 = x_0 - \lambda \frac{\partial J(x_0)}{\partial x} = 3 - .25(2*3-4) = 2.5$$
$$x_2 = x_1 - \lambda \frac{\partial J(x_1)}{\partial x} = 2.5 - .25(2*2.5-4) = 2.25$$
$$x_3 = x_2 - \lambda \frac{\partial J(x_2)}{\partial x} = 2.25 - .25(2*2.25-4) = 2.125$$

Figure1 illustrates the episodic process.

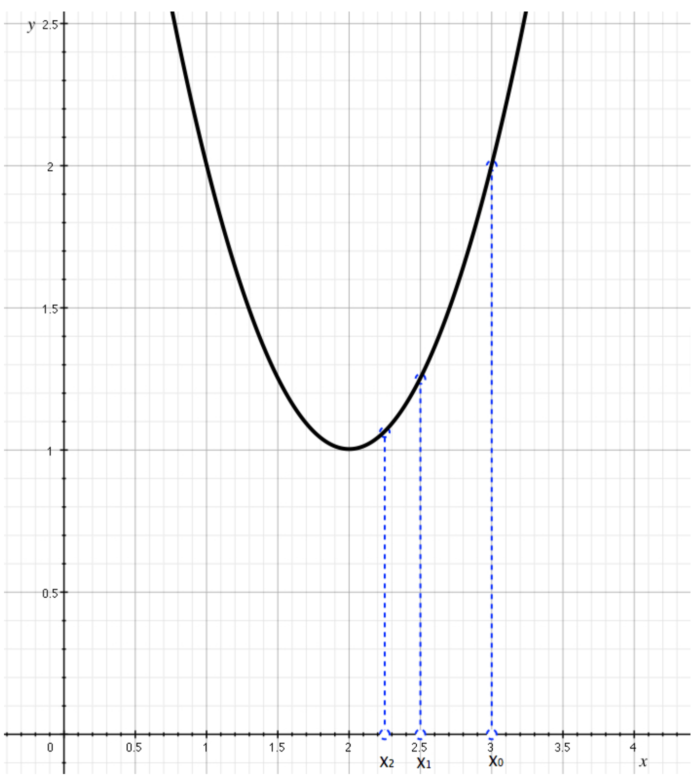

Figure 1: Gradient descent iterations